NASA Contractor Report 195292

High-Performance Parallel Analysis of Coupled Problems for Aircraft Propulsion

C.A. Felippa, C. Farhat, S. Lanteri, U. Gumaste, and M. Ronaghi
University of Colorado
Boulder, Colorado

March 1994

Prepared for
Lewis Research Center
Under Grant NAG3–1273

National Aeronautics and
Space Administration

Bi-Annual Report to

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center

Grant NAG3-1273

HIGH-PERFORMANCE PARALLEL ANALYSIS OF COUPLED PROBLEMS FOR AIRCRAFT PROPULSION

by

C. A. FELIPPA, C. FARHAT, S. LANTERI,

U. GUMASTE AND M. RONAGHI

Department of Aerospace Engineering Sciences and
Center for Space Structures and Controls
University of Colorado
Boulder, Colorado 80309-0429

June 1993

Report No. CU-CSSC-93-16

Summary

This research program deals with the application of high-performance computing methods for the analysis of complete jet engines. We have initited this program by applying the two-dimensional parallel aeroelastic codes to the interior gas flow problem of a by-pass jet engine. The fluid mesh generation, domain decomposition and solution capabilities were successfully tested.

1. Introduction

The present program deals with the application of high-performance parallel computation for the analysis of complete jet engines, considering the interaction of fluid, thermal and mechanical components. The research is driven by the simulation of advanced aircraft propulsion systems, which is a problem of primary interest to NASA Lewis.

The coupled problem involves interaction of structures with gas dynamics, heat conduction and heat transfer in aircraft engines. The *methodology* issues to be addressed include: consistent discrete formulation of coupled problems with emphasis on coupling phenomena; effect of partitioning strategies, augmentation and temporal solution procedures; sensitivity of response to problem parameters; and methods for interfacing multiscale discretizations in different single fields. The *computer implementation* issues to be addressed include: parallel treatment of coupled systems; domain decomposition and mesh partitioning strategies; data representation in object-oriented form and mapping to hardware driven representation, and tradeoff studies between partitioning schemes and fully coupled treatment.

2. Graduate Students

Two Ph. D. graduate students begin work this summer under support from the grant.

M. Ronaghi (U.S. citizen) began his doctoral studies at Colorado on January 1993. Mr. Ronaghi has a M.S. in Mechanical Engineering at North Carolina A&T State University at Greensbroro and has worked at NASA Langley doing finite element structural analysis. He has a good understanding of structures and composites and some computer experience but lacks background in fluid mechanics, thermomechanics and propulsion. He will remedy that by initiating a fluid course sequence this Spring semester and will start a thermal-propulsion sequence in the Fall semester.

U. Gumaste (permanent U.S. resident) begins his graduate studies at Colorado in the Fall semester, but will work on this project during June-July 1993 as an hourly R.A. Mr. Gumaste has a B.Tech in Civil Engineering from the Indian Institute of Technology, Bombay, India.

Both students were significantly helped by a visiting Post-Doc, Stéphane Lanteri, during their first modeling assignment. Dr. Lanteri is affiliated with INRIA Antipolis. He is working with Charbel Farhat in the development of parallel finite-volume/element methods for 2D and 3D flow around aircrafts, and the analysis of nonlinear fluid-structure interaction for flutter and stall analysis.

3. Flow Analysis of a By-Pass Engine

The main first-year objective is to "turn inside out" our exterior-domain aeroelastic codes to fit the interior-flow engine problem. The codes are then run to assess their strength

and weaknesses in numerical analysis and capturing physical effects. Observed weaknesses are then addressed by a combination of methodology and modeling improvements.

The gas flow within an engine is very complex. It exhibits localization, vortices, sharp pressure gradients and thermal-combustion effects. Our approach is to incorporate gas flow and structural modeling common to the exterior problem, and then solicit the help of experts to deal with new effects such as compression, diffussion, mixing and combustion.

To initiate this program we chose a rather old Conway by-pass engine sketched in the textbook of Hesse and Mumford [1]. Figure 1 is a schematic diagram of the engine presented in Hesse-Mumford's Fig. 11.7.

The purpose of the first experiments were to verify if the aeroelastic codes could be gracefully adapted to confined gas flow. To play it safe we began with a two-dimensional model and used the engine structure essentially as a way to provide boundary conditions for the gas flow. Blades and combustion effects were ignored.

The rather complex boundary configuration provided a good test for the fluid mesh generator, which "triangulates" the gas domain. This generation was done by S. Lanteri, who is an expert in this subject.

The fluid meshes were treated with Farhat's domain decomposer program DOMDEC [2]. Meshes were partitioned into 8 domains. Figures 3, 4 and 5 show the decompositions produced by the Greedy, Recursive Graph Bisection (RGB) and Reverse Cuthill-McKee (RCM) algorithms, respectievly. Ideally each partition should be single-connected to minimize interface communications overhead in parallel machines. Given the complex configuration of the gas domain, satisfaction of this criterion is by no means obvious. It can be seen that RGB met the single-connectivity criterion, but that Greedy and RCM did not.

The theoretical and computational basis of the gas flow calculations are described in the Appendix reprint of an article by Farhat, Lanteri and Fezoui [3]. Computations based on Stokes flow were carried out without difficulties. Figures 6 and 7 shoe contour plots of pressures and density, respectively, for the steady state corresponding to a free-flow Mach number of 0.4. Figure 8 shows the velocity field.

4. Future Work

The key need is to introduce more physical effects in the gas flow, namely compression, diffusion and combustion. We need to decide whether to continue with a two-dimensional axisymmetric model with artifices to represent nonaxisymmetric devices, or to proceed to a "sector" three-dimensional model requiring tetrahedral meshes. We plan to consult with NASA Lewis experts as to the best way to proceed at this point. Dr. Russ Claus of NASA LeRC has offered to provide us a three-dimensional model of a more recent engine. Such a model could be used as Testbed for the next phases of this research program.

References

[1] W. J. Hesse and N. V. S. Mumford, *Jet Propulsion for Aerospace Applications*, 2nd ed., Pitnam Pubs., New York, N.Y., 1964, Chapter 11

[2] C. Farhat and H. D. Simon, TOPS-DOMDEC: A SoftwareTool for Mesh Partitioning and Parallel Processing, Report CU-CSSC-93-11, Center for Space Structures and Controls, University of Colorado, Boulder, CO, May 1993

[3] C. Farhat, L. Fezoui and S. Lanteri, Two-dimensional viscous flow computations on the connection machine: unstructured meshes, upwind schemes, and massively parallel computations, *Computer Methods in Applied Mechanics and Engineering*, Vol. 102, (1993), pp. 61–88

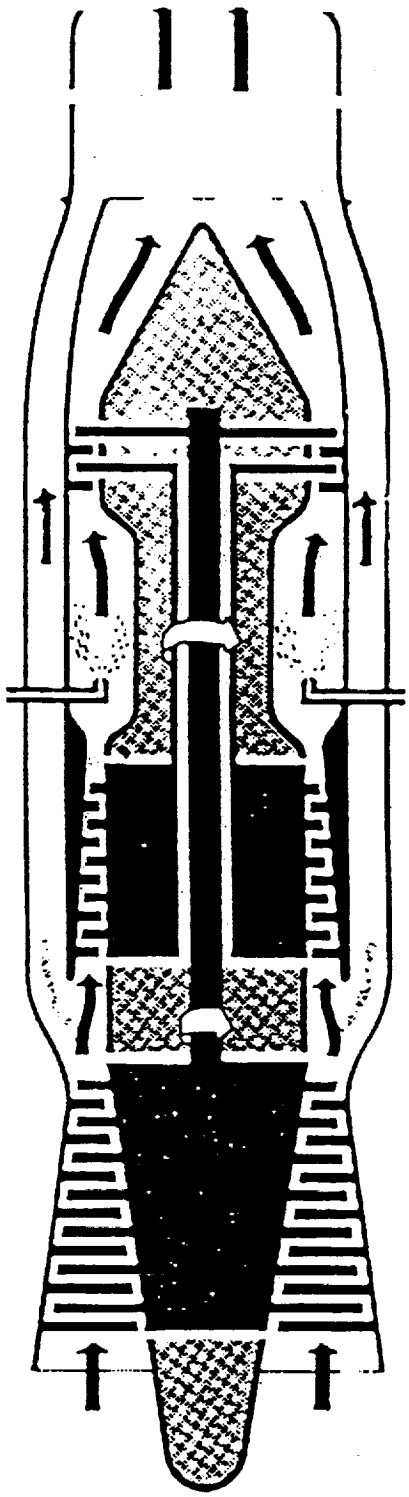

Figure 1 Schematics of Conway by-pass engine (from [1])

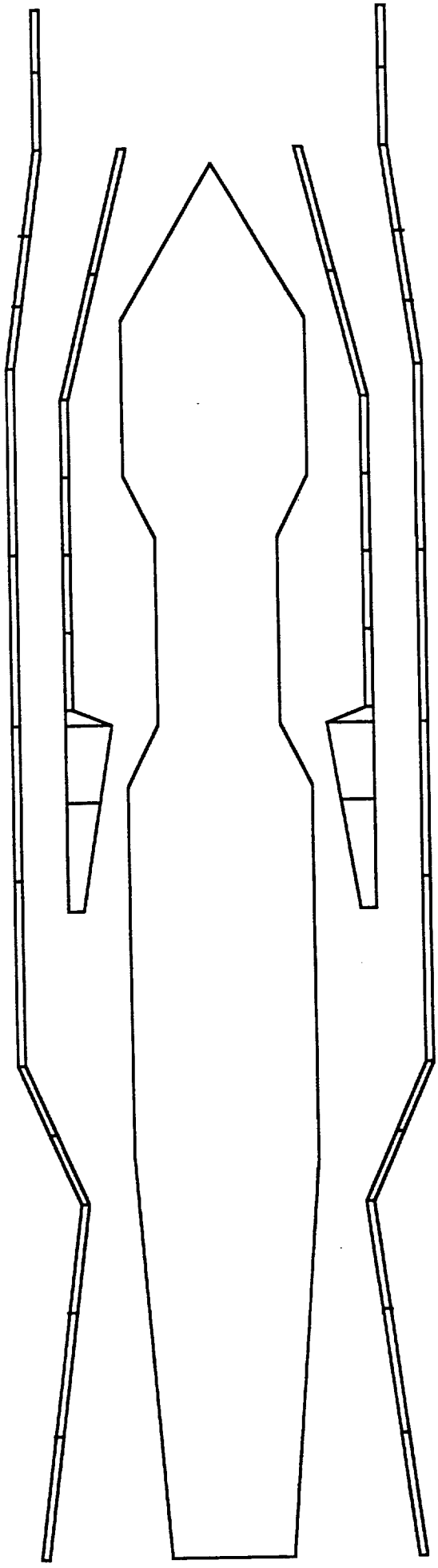

Figure 2. 2D Finite Element idealization of by-pass engine

Figure 3 Fluid mesh decomposition by Greedy algorithm implemented in DOMDEC

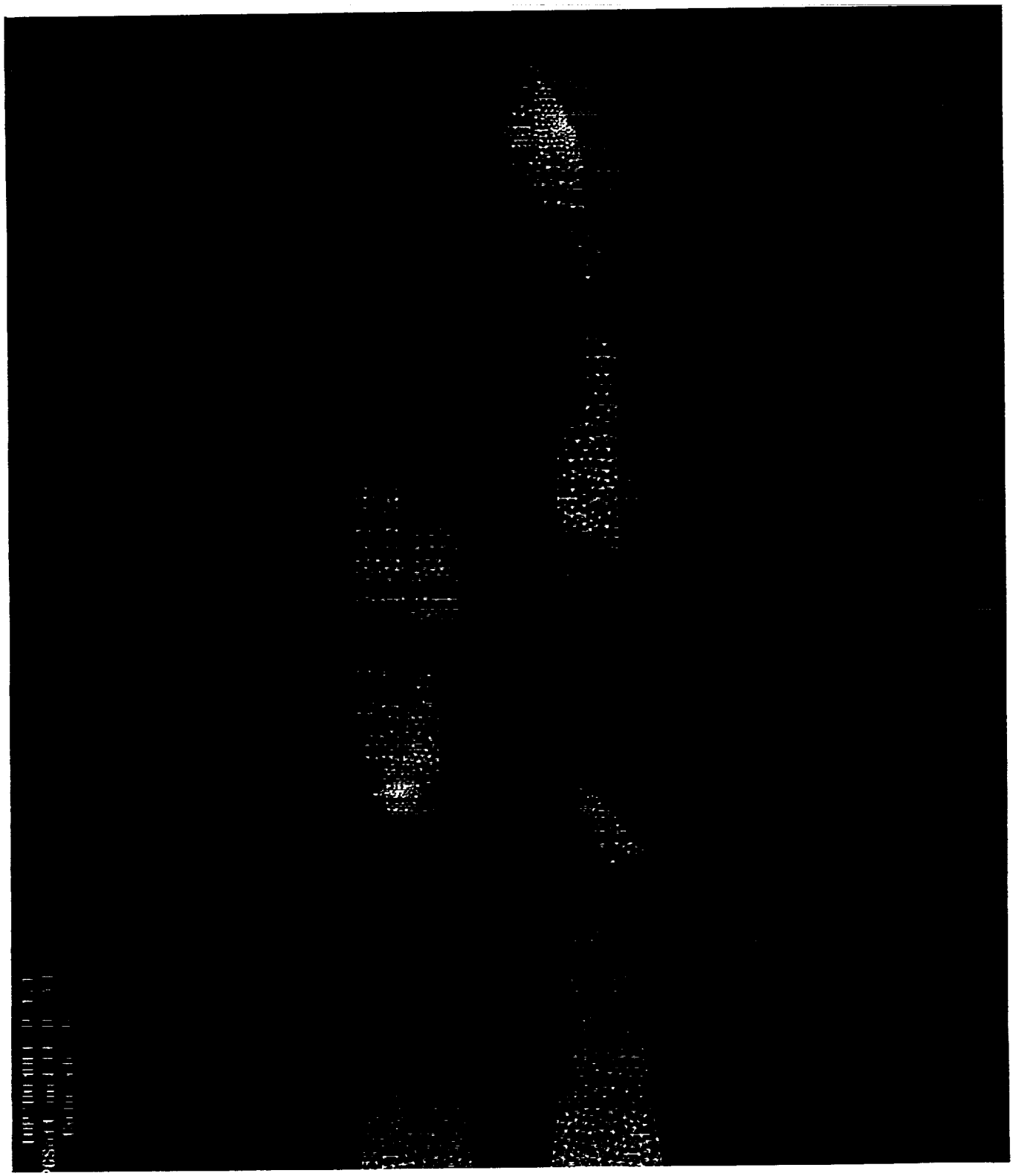

Figure 4 Fluid mesh decomposition by Recursive Graph Bisection (RGB) implemented in DOMDEC

Figure 5 Fluid mesh decomposition by Recursive Cuthill McKee (RCM) implemented in DOMDEC

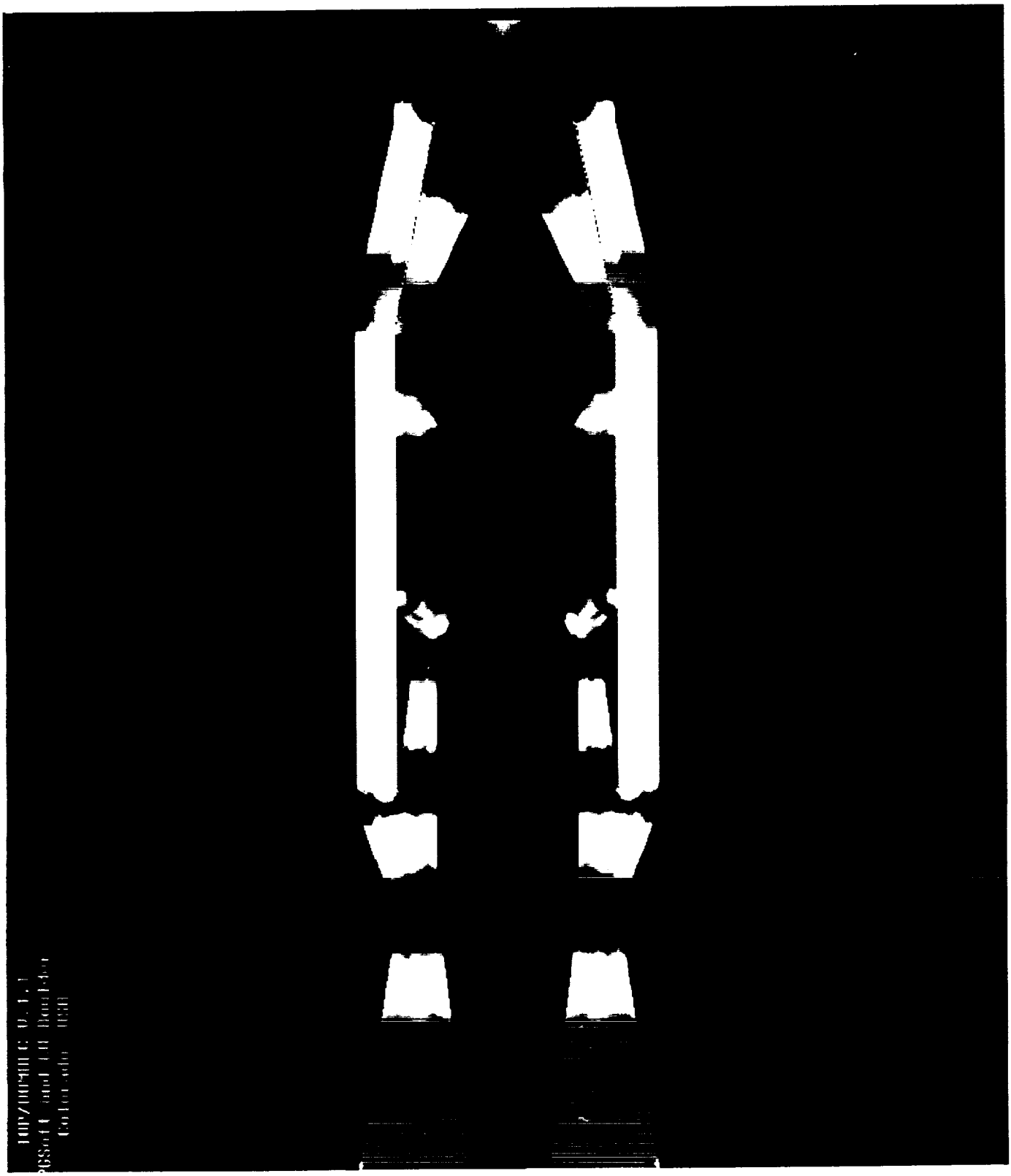

Figure 6 Computed steady-state pressure field

Figure 7 Computed steady-state density field

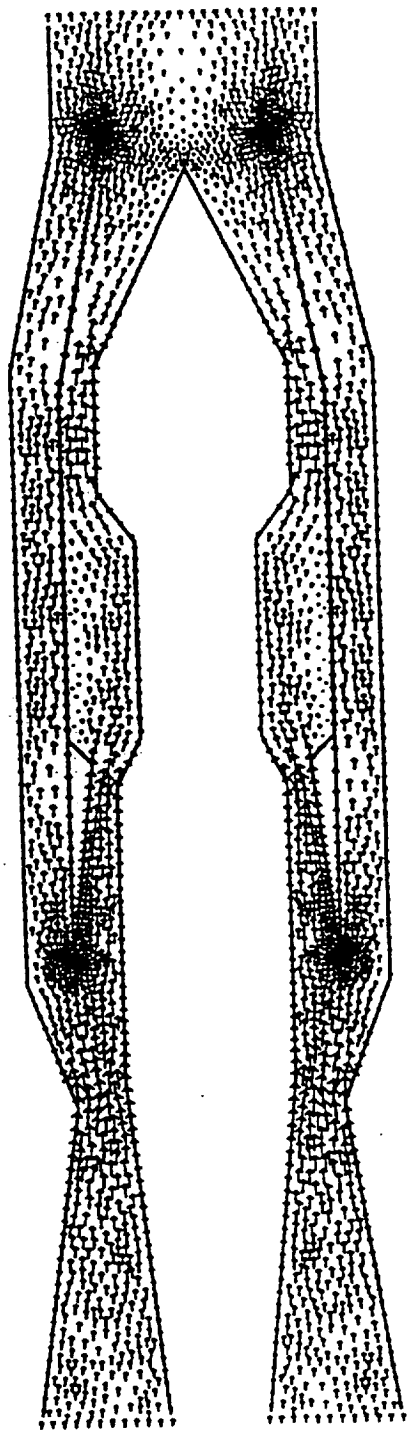

Figure 8 Computed steady-state velocity field

Appendix

Theoretical Background on Viscous Flow Computations

Summary

The following material, extracted from a recently published paper by Farhat, Fezoui and Lanteri [3], summarizes the theoretical foundations of our parallel Navier-Stokes computations on unstructured meshes. Although the article focuses on CM-2 computations carried out during 1990-1991, it also presents implementation considerations applicable to the present project.

1. Introduction

Previously we have reported on our experience with performing two-dimensional structured compressible flow computations on the Connection Machine CM-2 (Saati, Biringen and Farhat [A1], Lanteri, Farhat and Fezoui [A2]). We have found that this massively parallel processor is particularly well suited for explicit computations on regular grids. For grids that result in a high virtual processor ratio (VPR or VP ratio), using the NEWS fast communication mechanism, we have measured the communication component of the simulation time to represent typically less than 10% of the total CPU time. We have concluded that on a 64K machine (65536 processors), efficiency rates in the neighborhood of 2 gigaflops are attainable. We have also found that for both inviscid (Euler equations) and viscous (Navier-Stokes equations) flow structured computations, a 16K CM-2 (16384 processors) can be 4 and 6 times faster than one CRAY-2 processor, respectively.

We focus here on massively parallel viscous flow computations using fully unstructured grids. In Section 2, we formulate the problem to be solved, and in Section 3, we derive first-order and second-order spatial schemes that are characterized by an upwind integration of the convective fluxes. Second-order accuracy is achieved through a Monotonic Upwind Scheme for Conservation Laws (MUSCL) technique. An explicit, and therefore nicely parallelizable, Runge-Kutta method is selected for time integration; it is summarized in Section 4. Because the mesh irregularities inhibit the use of the NEWS mechanism, interprocessor communication is bound to be carried out via the slower machine router. If a trivial processor mapping is used, up to 60% of the total CPU time is consumed in communication requirements. This bottleneck has been previously analyzed and documented by Farhat, Sobh and Park [A3] for massively parallel finite element computations in solid mechanics problems. It has also been recently addressed by several other investigators for fluid flow computations. In particular, Shapiro [A4] has proposed the use of a graph coloring algorithm to allow a particular implementation of the communication steps which reduces the communication costs by a factor of two. Hammond and Barth [A5] have developed a vertex-based partitioning scheme for inviscid flow computations which attempts to minimize both the computational and communication costs associated with unstructured grids. Here, we present a strategy for mapping thousands of processors onto an unstructured grid which leads to an efficient scheme for carrying out communications of an arbitrary pattern. The key elements of this strategy are discussed in Section 5. These include the selection of an appropriate parallel data structure, the partitioning of a given unstructured grid into subgrids, and the mapping of each individual processor onto an entity of these subgrids. Combining this mapping strategy with a communication compiler reduces the communication overhead by an order of magnitude and brings it down to 15% of the total simulation time. In Section 6, we apply our massively parallel

code and its highly vectorized variant to the simulation of low Reynolds number chaotic flows. Measured performance results indicate that for such computations on unstructured grids, an 8K CM-2 with single precision floating point hardware is as fast as one CRAY-2 processor.

2. Mathematical modeling

First we recall the mathematical problem to be solved, and introduce the notation that is used in the sequel.

2.1. Governing equations

Let $\Omega \subset \Re^2$ be the flow domain of interest and Γ be its boundary. The conservative law form of the equations describing two-dimensional Navier-Stokes flows is given by :

$$\frac{\partial W}{\partial t} + \vec{\nabla}.\vec{\mathcal{F}}(W) = \frac{1}{Re}\vec{\nabla}.\vec{\mathcal{R}}(W) \tag{1}$$

where

$$\begin{aligned} W &= (\rho,\ \rho u,\ \rho v,\ E)^T \\ \vec{\nabla} &= \left(\frac{\partial}{\partial x},\ \frac{\partial}{\partial y}\right)^T \\ \vec{\mathcal{F}}(W) &= \begin{pmatrix} F(W) \\ G(W) \end{pmatrix} \\ \vec{\mathcal{R}}(W) &= \begin{pmatrix} R(W) \\ S(W) \end{pmatrix} \end{aligned} \tag{2}$$

The functions $F(W)$ and $G(W)$, and $R(W)$ and $S(W)$, denote the convective and diffusive fluxes, respectively. They can be written as :

$$\begin{aligned} F(W) &= \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix} \\ G(W) &= \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} \\ R(W) &= \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + \frac{\gamma k}{Pr}\frac{\partial \varepsilon}{\partial x} \end{pmatrix} \\ S(W) &= \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + \frac{\gamma k}{Pr}\frac{\partial \varepsilon}{\partial y} \end{pmatrix} \end{aligned} \tag{3}$$

where ρ is the density, $\vec{U} = (u,v)$ is the velocity vector, E is the total energy per unit of volume, p is the pressure, and ε is the specific internal energy. The variables p, E, ρ, $\vec{U}$, ε, and the temperature T are related by the state equation for a perfect gas:

$$p = (\gamma - 1)(E - \frac{1}{2}\rho \| \vec{U} \|^2) \tag{4}$$

and by:

$$\varepsilon = C_v T = \frac{E}{\rho} - \frac{1}{2}(\| \vec{U} \|^2) \tag{5}$$

where γ denotes the ratio of specific heats.

The components of the Cauchy stress tensor τ_{xx}, τ_{xy} and τ_{yy} are given by:

$$\tau_{xx} = \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) \qquad \tau_{yy} = \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) \qquad \tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{6}$$

where μ and k are the normalized viscosity and thermal conductivity coefficients. Two characteristic numbers appear in the above equations; the Reynolds number $Re = \frac{\rho_0 U_0 L_0}{\mu_0}$ where ρ_0, U_0, L_0 and μ_0 denote respectively, the characteristic density, velocity, length and diffusivity of the flow under consideration, and the Prandtl number $Pr = \frac{\mu_0 C_p}{k_0}$.

We consider the initial and boundary value problem (IBVP):

$$\begin{cases} \dfrac{\partial W}{\partial t} + \vec{\nabla}.\vec{\mathcal{F}}(W) = \dfrac{1}{Re}\vec{\nabla}.\vec{\mathcal{R}}(W) & \left(\vec{X},t\right) \in \Omega \times \Re^+ \\ W\left(\vec{X},0\right) = W_0(\vec{X}) & \vec{X} \in \Omega \\ W\left(\vec{X},t\right) = W_\Gamma(\vec{X}) & \vec{X} \in \Gamma = \partial\Omega \end{cases} \tag{7}$$

where W_0 and W_Γ are specified functions, and focus on finding a weak solution of (7) that is amenable to massively parallel computations.

2.2. Boundary conditions

We are mostly interested in external flows around airfoils. Therefore, we consider the case where the computational domain Ω is delimited by the boundary $\Gamma = \Gamma_b \cup \Gamma_\infty$. We denote by $\vec{\nu}$ the

outward unit normal at a given point of Γ (Fig. A1).

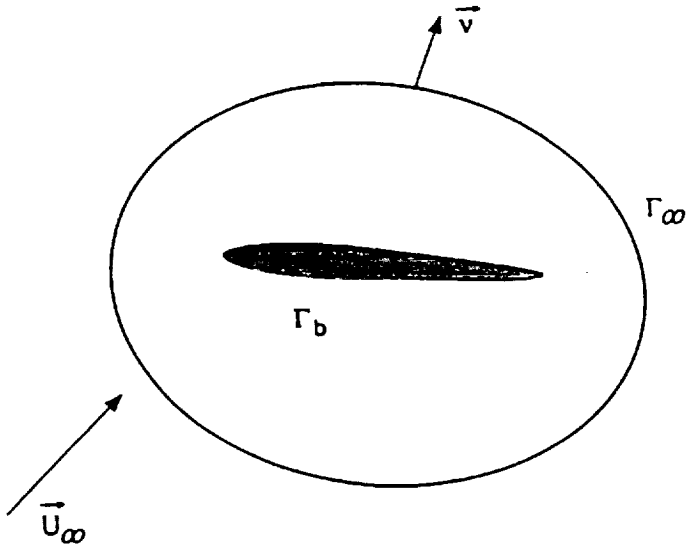

Fig. A1. *The computational domain*

In the far field, we assume that the viscous effects are negligible so that the flow is uniform. We adopt a formulation where the physical variables are non-dimensionalized. The free-stream vector W_∞ is given by:

$$\rho_\infty = 1 \quad \vec{U}_\infty = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \quad p_\infty = \frac{1}{\gamma M_\infty^2} \tag{8}$$

where α is the angle of attack and M_∞ is the free-stream Mach number. On the wall boundary Γ_b, we impose the no-slip condition and specify the temperature:

$$\vec{U} = \vec{0} \qquad T = T_b \tag{9}$$

We do not impose any boundary condition on the density. Therefore, the total energy per unit of volume and the pressure on the wall are given by :

$$E = \rho C_v T_b \qquad p = (\gamma - 1)E \tag{10}$$

3. Spatial discretization

3.1. Preliminary

The flow domain Ω is assumed to be a polygonal bounded region of $\Re^2$. Let $\mathcal{T}_h$ be a standard triangulation of Ω, and h the maximal length of the edges of $\mathcal{T}_h$. A vertex of a triangle Δ is denoted by S_i, and the set of its neighboring vertices by $K(i)$. At each vertex S_i, a cell C_i is constructed as the union of the subtriangles resulting from the subdivision by means of the medians of each triangle of $\mathcal{T}_h$ that is connected to S_i (Fig. A2). The boundary of C_i is denoted by ∂C_i, and the unit vector of the outward normal to ∂C_i by $\vec{\nu}_i = (\nu_{ix}, \nu_{iy})$. The union of all of the constructed cells forms a non-overlapping partition of the domain Ω:

$$\Omega = \bigcup_{i=1}^{ns} C_i \tag{11}$$

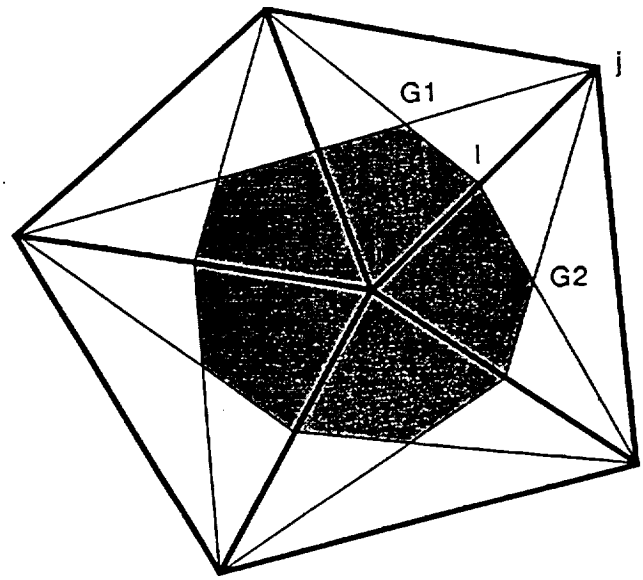

Fig. A2. *Cell definition in an unstructured grid*

For each cell C_i, a characteristic function Ψ_i is defined as :

$$\Psi_i(\vec{X}) = \begin{cases} 1 & \text{if } \vec{X} \in C_i \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Also, the following discrete spaces are introduced:

$$\begin{aligned} \mathcal{V}_h &= \{v_h \mid v_h \in C^0(\Omega),\ v_h|_\Delta \in P_1,\ \forall \Delta \in \mathcal{T}_h\} \\ \mathcal{W}_h &= \{v_h \mid v_h \in L^2(\Omega),\ v_h|_{C_i} = v_i = \text{constant},\ i = 1, ..., ns\} \end{aligned} \tag{13}$$

where P_1 is the space of polynomials in two variables and of degree 1. Clearly, any function f belonging to $\mathcal{V}_h$ is uniquely determined by its values $f(S_i)$ at each vertex S_i, and can be expressed as:

$$f(\overrightarrow{X}) = \sum_{i=1,ns} f(S_i) N_i(\overrightarrow{X}) \tag{14}$$

where $\{N_i\}_{i=1}^{i=ns}$ is a basis of $\mathcal{V}_h$. Finally, it is noted that a natural bijection between the spaces $\mathcal{V}_h$ and $\mathcal{W}_h$ can be constructed as:

$$\forall f \in \mathcal{V}_h \quad , \quad S(f(\overrightarrow{X})) = \sum_{i=1,ns} f(S_i) \Psi_i(\overrightarrow{X}) \tag{15}$$

3.2 Variational formulation and first order spatial approximations

A variational formulation of the IBVP (7) goes as follows:

$$\text{Find } W_h \in (\mathcal{V}_h)^4, \ \forall \varphi_h \in \mathcal{V}_h$$

$$\begin{aligned} \int_\Omega \frac{\partial W_h}{\partial t} \varphi_h dx dy + \int_\Omega \overrightarrow{\nabla} . \overrightarrow{\mathcal{F}}(W_h) \varphi_h dx dy \\ = \frac{1}{Re} \int_\Omega \overrightarrow{\nabla} . \overrightarrow{\mathcal{R}}(W_h) \varphi_h dx dy \end{aligned} \tag{16}$$

We construct a mixed finite volume/finite element (Galerkin) approximation for solving the above problem by introducing appropriate schemes for computing the left and right-hand-side integrals of (16). Chosing φ_h as the shape function N_i associated with the node S_i and applying the operator S to the left hand side of (16) leads to a mass-lumped variational approach which transforms the above equation into:

$$\begin{aligned} \int_{C_i} \frac{\partial W_h}{\partial t} dx dy + \int_{C_i} \overrightarrow{\nabla} . \overrightarrow{\mathcal{F}}(W_h) dx dy \\ = \frac{1}{Re} \int_{SupN_i} \overrightarrow{\nabla} . \overrightarrow{\mathcal{R}}(W_h) N_i dx dy \end{aligned} \tag{17}$$

where $SupN_i = \bigcup_{\Delta, S_i \in \Delta} \Delta$. Using Green's formula for the convective term and integrating by part the diffusive one leads to:

$$\int_{C_i} \frac{\partial W_h}{\partial t} dxdy + \int_{\partial C_i} \vec{\mathcal{F}}(W_h).\vec{\nu}_i d\sigma$$
$$= -\frac{1}{Re} \sum_{\Delta, S_i \in \Delta} \int_{\Delta} \vec{\mathcal{R}}(W_h).\overrightarrow{\nabla N_i^{\Delta}} dxdy \qquad (18)$$
$$+ \frac{1}{Re} \int_{\Gamma_b \cup \Gamma_\infty} \vec{\mathcal{R}}(W_h).\vec{\nu}_i N_i d\sigma$$

where N_i^{Δ} is the restriction of N_i to triangle Δ. Finally, we drop the right hand side boundary integral as we enforce the viscous boundary conditions in a strong form on Γ_b and neglect the viscous effects on Γ_∞, so that equation (18) simplifies to:

$$\int_{C_i} \frac{\partial W_h}{\partial t} dxdy + \sum_{j \in K(i)} \int_{\partial C_{ij}} \vec{\mathcal{F}}(W_h).\vec{\nu}_{ij} d\sigma < 1 >$$
$$+ \int_{\partial C_i \cap \Gamma_b} \vec{\mathcal{F}}(\overline{W}_h).\vec{\nu}_i d\sigma < 2 >$$
$$+ \int_{\partial C_i \cap \Gamma_\infty} \vec{\mathcal{F}}(\overline{W}_h).\vec{\nu}_i d\sigma < 3 > \qquad (19)$$
$$= - - \frac{1}{Re} \sum_{\Delta, S_i \in \Delta} \int_{\Delta} \vec{\mathcal{R}}(W_h).\overrightarrow{\nabla N_i^{\Delta}} dxdy < 4 >$$

where $\overline{W}_h$ is the specified value of W_h at the boundaries.

The reader should note that the above formulation leads to a locally one-dimensional computation of each convective term, along the normal direction $\vec{\nu}$. For this purpose, the boundary ∂C_i of the cell C_i is split into bi-segments ∂C_{ij} which join the middle point of the edge $[S_i S_j]$ to the centroids of the triangles having both of S_i and S_j as vertices (Fig. A3), and the integral $< 1 >$ is evaluated as:

$$\sum_{j \in K(i)} \int_{\partial C_{ij}} \vec{\mathcal{F}}(W_h).\vec{\nu}_{ij} d\sigma = \sum_{j \in K(i)} \vec{\mathcal{F}}(\tilde{U}). \int_{\partial C_{ij}} \vec{\nu}_{ij} d\sigma \qquad (20)$$

where $\vec{\mathcal{F}}(\tilde{U})$ is some approximation of the convective flux computed at the interface between cells C_i and C_j.

Following Fezoui and Stoufflet [A6], we choose $\vec{\mathcal{F}}(\tilde{U})$ to be a numerical flux function Φ associated with a first-order accurate upwind scheme (Van Leer [A7]). It is denoted here by $H_{ij}^{(1)}$, where the superscript $^{(1)}$ emphasizes the first order accuracy, and can be written as:

$$H_{ij}^{(1)} = \Phi_{\mathcal{F}_{ij}}(W_i, W_j, \vec{\nu}_{ij}) \qquad (21)$$

where $W_i = W_h(S_i)$ and $W_j = W_h(S_j)$. For example, the following numerical flux functions can be used to construct $H_{ij}^{(1)}$:

- *Roe's Scheme* [A8]

$$\Phi_{\mathcal{F}}^{R}\left(U,V,\vec{\nu}\right) = \frac{\mathcal{F}(U,\vec{\nu}) + \mathcal{F}(V,\vec{\nu})}{2} - d(U,V,\vec{\nu}) \tag{22}$$

where $d\left(U,V,\vec{\nu}\right)$ is a numerical diffusivity defined as:

$$d\left(U,V,\vec{\nu}\right) = \mid \mathcal{A}\cdot\left(\tilde{W},\vec{\nu}\right) \mid \frac{(V-U)}{2} \tag{23}$$

and $\tilde{W}$ is some mean value of U et V.

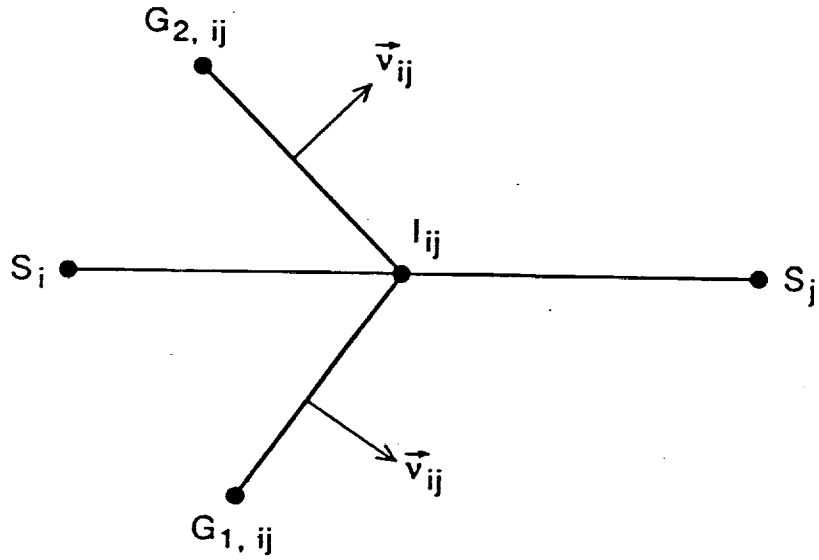

Fig. A3. *Splitting of* ∂C_{ij}

- *Steger and Warming's scheme* [A9]

$$\Phi_{\mathcal{F}}^{SW}\left(U,V,\vec{\nu}\right) = \mathcal{A}^{+}\left(U,\vec{\nu}\right)U + \mathcal{A}^{-}\left(V,\vec{\nu}\right)V \tag{24}$$

where $\mathcal{A} = \mathcal{A}^{+} + \mathcal{A}^{-}$ and $\mid \mathcal{A} \mid = \mathcal{A}^{+} - \mathcal{A}^{-}$.

The viscous integral $< 4 >$ is evaluated via a classical Galerkin finite element $P1$ method which results in a centered scheme. Since the approximations of the physical variables are taken

in $\mathcal{V}_h$, the components of the stress tensor and those of $\overrightarrow{\nabla N_i^\Delta}$ are constant in each triangle. The velocity vector in a triangle is computed as:

$$\overrightarrow{U}_\Delta = \frac{1}{3} \sum_{k=1, k\epsilon\Delta}^{3} \overrightarrow{U}^k \tag{25}$$

Consequently, the viscous fluxes are evaluated as:

$$\sum_{\Delta, S_i \in \Delta} \int_\Delta \overrightarrow{\mathcal{R}}(W_h).\overrightarrow{\nabla N_i^\Delta} dxdy = \sum_{\Delta, S_i \in \Delta} area(\Delta) \left(R_\Delta \frac{\partial N_i^\Delta}{\partial x} + S_\Delta \frac{\partial N_i^\Delta}{\partial y} \right) \tag{26}$$

where R_Δ and S_Δ are the constant values of $R(W)$ and $S(W)$ in the triangle Δ.

3.3. Higher order extension

The numerical integration with an upwind scheme described above leads to a spatial approximation that is only first-order accurate. Here, we focus on constructing a second-order accurate solution without changing the space of approximations. We develop a second-order scheme that is an extension of Van Leer's MUSCL method [A7] to the case of unstructured meshes.

Usually, a second-order approximation requires the evaluation of the gradient of the solution at each vertex. Clearly, the gradient of a function v_h of $\mathcal{V}_h$ is constant in each element and discontinuous in the flow domain. Following the MUSCL method, one way to achieve second-order spatial accuracy is to evaluate the fluxes with extrapolated values W_{ij}, W_{ji} at the interface $\partial C_i \cap \partial C_j$. Basically, this leads to substituting $H_{ij}^{(1)}$ in the previous scheme by $H_{ij}^{(2)}$ which is given by:

$$\begin{aligned} H_{ij}^{(2)} &= \Phi_{\mathcal{F}_{ij}}(W_{ij}, W_{ji}, \overrightarrow{\nu}_{ij}) \\ W_{ij} &= W_i + \tfrac{1}{2}(\overrightarrow{\nabla} W)_i^\beta . \overrightarrow{S_i S_j} \\ W_{ji} &= W_j - \tfrac{1}{2}(\overrightarrow{\nabla} W)_j^\beta . \overrightarrow{S_i S_j} \end{aligned} \tag{27}$$

where the approximate nodal gradients $(\overrightarrow{\nabla} W)_{i,j}^\beta$ are obtained via a β-combination of centered and fully upwind gradients :

$$(\overrightarrow{\nabla W})_i^\beta = (1-\beta)(\overrightarrow{\nabla W})_i^{Cent} + \beta(\overrightarrow{\nabla W})_i^{Upw} \tag{28}$$

Here, a centered gradient $(\overrightarrow{\nabla W})_i^{Cent} = (\overrightarrow{\nabla W})^{\beta=0}$ can be chosen as any vector satisfying:

$$(\overrightarrow{\nabla W})_i^{Cent} . \overrightarrow{S_i S_j} = W_j - W_i \tag{29}$$

A nicely parallelizable scheme for computing the upwind gradients $(\overrightarrow{\nabla W})_i^{Upw}$ goes as follows. First, we note that $(\overrightarrow{\nabla W})_i^{Upw} = (\overrightarrow{\nabla W})^{\beta=1}$, and from (28) we derive:

$$(\overrightarrow{\nabla W})_i^{Upw} = 2(\overrightarrow{\nabla W})_i^{\beta=\frac{1}{2}} - (\overrightarrow{\nabla W})_i^{Cent} \tag{30}$$

We compute the half-upwind gradients ($\beta = \frac{1}{2}$) via a linear interpolation of the Galerkin gradients computed in each triangle of C_i, so that:

$$\begin{aligned}\left(\overrightarrow{\nabla W}\right)_i^{\beta=\frac{1}{2}} &= \frac{\int_{C_i} \overrightarrow{\nabla W}|_{\Delta}\, dxdy}{\int_{C_i} dxdy} \\ &= \frac{1}{area(C_i)} \sum_{\Delta \epsilon C_i} \frac{area(T)}{3} \sum_{k=1,k\epsilon T}^{3} W^k \overrightarrow{\nabla \varphi_k}\end{aligned} \tag{31}$$

Finally, we evaluate the nodal gradients using the following third-order biased scheme:

$$\begin{aligned}(\overrightarrow{\nabla W})_i^{\beta=\frac{1}{3}} &= \frac{2}{3}(\overrightarrow{\nabla W})_i^{\beta=0} + \frac{1}{3}(\overrightarrow{\nabla W})_i^{\beta=1} \\ &= \frac{2}{3}(\overrightarrow{\nabla W})_i^{\beta=0} + \frac{1}{3}\left(2(\overrightarrow{\nabla W})_i^{\beta=\frac{1}{2}} - (\overrightarrow{\nabla W})_i^{\beta=0}\right) \\ &= \frac{1}{3}(\overrightarrow{\nabla W})_i^{\beta=0} + \frac{2}{3}(\overrightarrow{\nabla W})_i^{\beta=\frac{1}{2}}\end{aligned} \tag{32}$$

3.4. Boundary conditions

The second term $< 2 >$ and the third term $< 3 >$ of the right-hand side of (19) contain the physical boundary conditions. These are represented by the vector $\overline{W}_h$ which involves quantities that depend on the interior values of W_h, and quantities that are determined by the physical boundary conditions.

Wall boundary : the no-slip condition is enforced in a strong form (9, 10) so that the corresponding boundary integral $< 2 >$ does not need to be evaluated.

Inflow and outflow boundaries : at these boundaries, a precise set of compatible exterior data which depend on the flow regime and the velocity direction must be specified. For that purpose, a *plus-minus* flux splitting is applied between exterior data and interior values. More precisely, the boundary integral $< 3 >$ is evaluated using a non-reflective version of the flux-splitting of Steger and Warming [A9] :

$$\int_{\partial C_i \cap \Gamma_\infty} \mathcal{F}(\overline{W}_h).\overrightarrow{\nu}_i d\sigma = \mathcal{A}^+(W_i, \overrightarrow{\nu}_{i\infty}).W_i + \mathcal{A}^-(W_i, \overrightarrow{\nu}_{i\infty}).W_\infty \tag{33}$$

4. Time discretization

The resulting semi-discrete fluid flow equations can be written as:

$$\frac{dW}{dt} + \psi(W) = 0 \tag{34}$$

Because it lends itself to massive parallelism, the explicit Runge-Kutta method is selected for integrating the above equations. A 3-step variant is used here. It is summarized as :

$$\begin{cases} W^{(0)} = W^n \\ W^{(k)} = W^{(0)} - \dfrac{\Delta t}{4-k}\psi(W^{(k-1)}) \qquad k = 1,2,3 \\ W^{n+1} = W^{(3)} \end{cases} \tag{35}$$

The above scheme is often referred to as the low-storage Runge-Kutta method as only the solution at substep $\alpha - 1$ is used to compute the one at substep α. It is third-order accurate in the linear case, but only second-order accurate in our case.

5. Parallel implementation on the Connection Machine CM-2

Clearly, expressions (19) and (27-35) reveal that both the spatial and temporal integrations are in principle nicely parallelizable. In this section, our interest lies in investigating the most efficient way to implement these computations on a Single Instruction Multiple Data (SIMD) massively parallel computer such as the Connection Machine CM-2. Special care is given to interprocessor communication because mesh irregularities: (a) inhibit the exploitation of the NEWS grid, so that the relatively slow router must be used, and (b) induce a different amount of communication steps within each processor, which is not particularly desirable on a SIMD machine. Rather than overviewing the CM-2, we refer the reader to the technical summary of Thinking Machines [A10] for architectural details, and to Farhat, Sobh, and Park [A3] for an in-depth analysis of interprocessor communication on the CM-2 when computing over an irregular mesh.

5.1. Parallel data structure

Behind the performance of any parallel algorithm lies the choice of the corresponding parallel data structure. The latter is closely related to both the entity and the task to be assigned to each processor. Therefore, all of the computational, communication and memory requirements should be considered before the distributed data structure is determined. For the mixed finite volume/finite element method presented here, we consider four candidates for a fundamental entity (Fig. A4):

- the vertex S_i,
- the edge E_{ij} joining the vertices S_i and S_j,
- the element (here the triangle) Δ_{ijk} connecting the vertices S_i, S_j and S_k,

- and the cell C_i defined in Section 3.1.

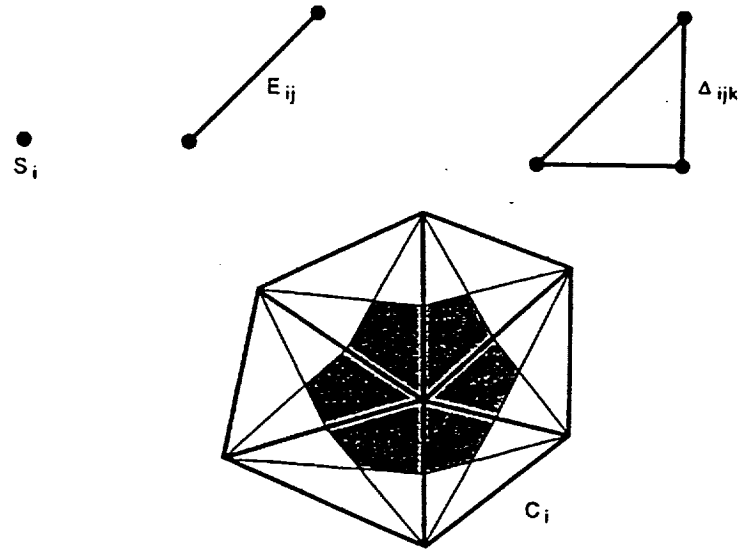

Fig. A4. *Fundamental entity candidates*

Memory considerations

While regular grids are most often characterized (in terms of memory requirements) by their number of vertices N_V, irregular triangular grids can be also characterized by either their number of elements N_Δ, or by their number of edges N_E. Here, we assume for simplicity that $\mathcal{T}_h$ is characterized by its number of vertices. Euler's relations for a triangulation state that :

$$\begin{aligned} N_V + N_\Delta - N_E &= 1 \\ 2N_E - N_{BV} &= 3N_\Delta \end{aligned} \tag{36}$$

where N_{BV} denotes the number of vertices at the boundary of the triangulation. This implies that :

$$N_\Delta \approx 2N_V \quad \text{and} \quad N_E \approx 3N_V \tag{37}$$

Therefore, if $\mathcal{T}_h$ is designed, for example, so that its number of vertices matches a given Connection Machine size, the VP ratio associated with each data structure candidate varies as indicated below:

	Vertex	Edge	Element	Cell
VPR	1	3	2	1

The reader should note that for the edge case, the machine automatically selects a VP ratio of 4, since it is the closest power of two to the theoretical VPR. Clearly, the vertex and cell entities are the best candidates on the sole basis of efficient memory usage.

Operation count

The numerical algorithms discussed in Section 2 and Section 3 can be organized around three basic computational steps :

(Step a) evaluation of the Galerkin gradients (32),
(Step b) evaluation of the diffusive fluxes (26),
(Step c) and evaluation of the convective fluxes (27).

While **Step (c)** is most efficiently performed using edge-wise computations, **Step (a)** and **Step (b)** are inherently element-level calculations. Therefore, whatever fundamental entity is selected, it must contain both edge and element information, which rules out the edge E_{ij} data structure.

On the other hand in an element-based partition, every triangle Δ_{ijk} provides direct access to all of the three edges E_{ij}, E_{jk} and E_{ki}. However in that case, two VP sets must be used; one containing N_Δ processors which store triangle related data (geometrical data), and another one containing N_V processors which store vertex related data (physical data). Otherwise, if only one set of virtual processors is used and assigned to both triangle and vertex data, a nodal result would be duplicated in as many processors as there are triangles connected to that vertex.

The vertex entity S_i is an effective candidate only when augmented with the auxiliary data structures that can handle the data associated with the elements and edges connected to a given vertex — that is, when transformed into a cell data structure.

Finally, we note that the cell entity stores both vertex and element data, and therefore provides access to all of vertex, element and edge information. Consequently, only element and cell partitions are retained for further discussions.

Next, we evaluate the operation count for each of **Step (a)**, **Step (b)** and **Step (c)**, assuming an element- or cell-based data structure. We denote by C_c^E and C_{ab}^Δ, the number of arithmetic operations associated with one edge computation during **Step (c)**, and with one triangle computation during **Step (a)** and **Step (b)**, respectively. The computational complexities characterizing the two retained candidates are tabulated below.

	Element	**Cell**
Step (c)	$2 \times C_c^E$	$2 \times C_c^E$
Step (a) + Step (b)	C_{ab}^Δ	$3 \times C_{ab}^\Delta$

In both an element- and cell-based partition, an edge is shared by two virtual processors, so that the flux $H_{ij}^{(2)}$ across $[S_iS_j]$ is computed twice. Only an edge partition would eliminate these redundant computations, but that choice has already been eliminated. In a cell-based partition, a triangle Δ_{ijk} is shared by three virtual processors, and therefore additional redundant computations are generated.

Communication costs

The computational steps discussed above require four communication steps denoted here by **(c1)**, **(c2)**, **(c3)**, and **(c4)**. These are discussed below for the element and cell parallel data structures.

First, we consider the case of an element-based partition. During the first communication step (**c1**), each virtual processor assigned to a triangle Δ_{ijk} gets the physical states at vertices S_i, S_j and S_k from neighboring processors. Then, the computations in **Step (a)** and **Step (b)** are carried out. During the second communication step (**c2**), the element-wise results are sent back to the virtual processors holding vertex data. The latter virtual processors use these values to compute the nodal gradients (32) and diffusive fluxes (26). In step (**c3**) the nodal gradients are communicated to neighboring processors. Next, each virtual processor evaluates three second-order convective fluxes (15) across the three edges connected by triangle Δ_{ijk}. During the last communication step (**c4**), the edge-wise fluxes are sent to the virtual processors holding vertex data.

Communication with a cell-based partition is more complex, as each cell may have a different number of neighbors. However, fewer communication steps are needed because each virtual processor stores within its local memory all of the element-wise values that are necessary for the evaluation of the nodal gradients and the diffusive fluxes, as well as the elemental convective fluxes.

The communication count associated with the four steps (**c1**) to (**c4**) is tabulated below for each of the two retained data structure candidates. N_{neigh}^{max} denotes the maximum number of neighboring cells.

	Element	**Cell**
(c1)	3	N_{neigh}^{max}
(c2)	3	0
(c3)	3	N_{neigh}^{max}
(c4)	6	0

Selected candidate

The operation and communication counts are summarized below for both the element and cell data structures. Equations (36) are used to express the results in terms of the number of vertices in the mesh.

	Element	**Cell**
Operation count	$(6 \times C_c^E + 2 \times C_{ab}^{\Delta}) \times N_V$	$(6 \times C_c^E + 6 \times C_{ab}^{\Delta}) \times N_V$
Communication count	$30 \times N_V$	$12 \times N_V$

Clearly, redundant arithmetic operations can be avoided only at the expense of additional communication characterized by an irregular pattern, which is usually not beneficial on a massively parallel processor such as the CM-2. Therefore, we have chosen the cell-based parallel data structure and have accepted the additional cost of redundant flux computations. Hammond and Barth [A5] have invoked a graph theory result due to Chrobak and Eppstein [A17] to eliminate redundant edge-based flux computations for Euler flows. This result states that for any planar graph, there exists an orientation of the edges such that no vertex has more than three edges directed out from it. This means that there exists a cell partition where no processor needs to compute the convective fluxes across more than three edges of the computational cell. However, this graph theory result does not apply for our viscous computations because these also include element-based operations.

5.2. Grid decomposition and processor mapping

Efficiency in arbitrary communication on the CM-2 requires the minimization of both the "hammering" on the router — that is, wire contention, and the distance that information has to travel — that is, the number of hops between the sender and receiver processors. Here, this implies that : (a) adjacent cells must be assigned, as much as possible, to directly connected processors or processors that are lying in directly connected chips, and (b) contention for the wire connecting neighboring chips must be reduced.

In a first step, the unstructured grid is decomposed into a series of subgrids each containing 16 adjacent numerical cells. Each subgrid is assigned to a certain CM-2 chip that is subsequently identified, so that adjacent cells within a subgrid are assigned to directly connected processors lying in the same chip. As a result, off-chip communication is needed only across the subgrid boundaries. Wire contention is reduced if each of the defined subgrids is surrounded by the largest possible number of neighboring subgrids. Indeed, wherever a subgrid boundary is shared with several other subgrids, off-chip communication is split between distinct chips and is distributed across several of the available inter-chip wires (Fig. A5). On the other hand, if for example a subgrid is adjacent only to two other subgrids, a maximum of two wires can be used during off-chip communication, which may create a severe wire contention that would serialize communication and significantly increase its cost. Here, we use the mesh decomposer of Farhat [A11] which has proven to be very effective at reducing wire contention on the CM-2 (Farhat, Sobh and Park [A3]).

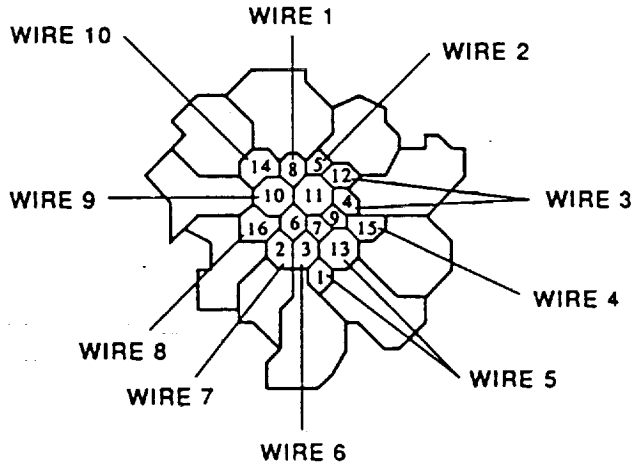

Fig. A5. *Grid decomposition with reduced wire-contention*

The next step is to reduce the distance that information has to travel during off-chip communication, that is when data is exchanged between centers of cells that are assigned to processors lying on different chips. This can be achieved by assigning adjacent subgrids as far as possible to directly connected chips. A combinatorial optimization-like procedure known as *Simulated An-*

nealing (see, for example, Flower, Otto and Salama [A12]) is probably the most popular technique for tackling this mapping problem. However, it is a very expensive procedure which has often proved to be impractical. Alternative heuristic-based schemes have been developed by several authors including Bokhari [A13], Farhat [A14], and recently Hammond and Schreiber [A15]. In this work, we have adopted the mapper of reference [A14]. It is based on a combined greedy/divide and conquer approach and is tuned for hypercube topologies.

A detailed analysis of interprocessor communication on the CM-2 for unstructured grids can be found in Farhat, Sobh and Park [A3]. In that reference, it is shown that mesh irregularities induce an MIMD (Multiple Instruction Multiple Data) style of programming for the communication phase which dominates the cost of communication. It is also suggested that since the irregular pattern of communication is fixed in time, a considerable improvement can be achieved if that pattern is evaluated during the first time step, then compiled or stored in the CM-2 for re-use in subsequent time steps. However, no software was available at that time for validating the proposed communication strategy. Recently, a communication compiler prototype has become available (Dahl [A16]) and can be used for storing the routing pattern. In Section 6, we report on its performance.

6. Numerical Experiments

(This Section reports on numerical experiments on the CM-2 and Cray 2. Since airfoil problems are of limited important for the present research, they are not presented here.)

7. Closure

Mixed finite volume/finite element spatial schemes for fully unstructured grids are developed and implemented on the CM-2, and applied to the simulation of two-dimensional viscous flows. Second-order accuracy in the discretization of the convective fluxes is achieved through a Monotonic Upwind Scheme for Conservation Laws (MUSCL) technique. The diffusive fluxes are computed using a classical Galerkin finite element method, and the resulting semi-discrete equations are time integrated with an explicit Runge-Kutta algorithm.

A strategy for mapping thousands of processors onto an unstructured grid is presented. Its key elements are given by the selection of an appropriate parallel data structure, the careful partitioning of a given unstructured grid into specific subgrids, and the mapping of each individual processor onto an entity of these subgrids. Whenever the communication patterns are compiled during the first time step, the total time elapsed in interprocessor communication using the router is drastically reduced to represent only 15% of the total CPU time of the simulation.

References

[A1] A. Saati, S. Biringen and C. Farhat, "Solving Navier-Stokes Equations on a Massively Parallel Processor: Beyond the One Gigaflop Performance," *Int. J. Supercomp. Appl.*, Vol. 4, No. 1, pp. 72-80, (1990).

[A2] S. Lanteri, C. Farhat and L. Fezoui, "Structured Compressible Flow Computations on the Connection Machine," *INRIA Report No. 1322*, (1990).

[A3] C. Farhat, N. Sobh and K. C. Park, "Transient Finite Element Computations on 65536 Processors : The Connection Machine," *Int. J. Num. Meth. Eng.*, Vol. 30, pp. 27-55, (1990).

[A4] R. A. Shapiro, "Implementation of an Euler/Navier-Stokes Finite Element Algorithm on the Connection Machine," *AIAA Paper 91-0483,* 29th Aerospace Sciences Meeting, Reno (1991).

[A5] S. Hammond and T. Barth, "An Efficient Massively Parallel Euler Solver for Unstructured Grids," *AIAA Paper 91-0441,* 29th Aerospace Sciences Meeting, Reno, Nevada (1991).

[A6] L. Fezoui and B. Stoufflet, "A Class of Implicit Upwind Schemes for Euler Simulations with Unstructured Meshes," *J. Comp. Phys.*, Vol. 84, pp. 174-206, (1989).

[A7] B. Van Leer, "Towards the Ultimate Conservative Difference Scheme V: a Second-Order Sequel to Goudonov's Method," *J. Comp. Phys.*, Vol. 32, (1979).

[A8] P. L. Roe, "Approximate Riemann Solvers, Parameters Vectors and Difference Schemes," *J. Comp. Phys.*, Vol. 43, pp. 357-371, (1981).

[A9] J. Steger and R. F. Warming, "Flux Vector Splitting for the Inviscid Gas Dynamic with Applications to Finite-Difference Methods," *J. Comp. Phys.*, Vol. 40, No. 2, pp. 263-293, (1981).

[A10] Thinking Machines Corporation, "Connection Machine Model CM-2: Technical Summary," Version 6.0, (1990).

[A11] C. Farhat, "A Simple and Efficient Automatic Finite Element Mesh Domain Decomposer," *Comp. & Struct.*, Vol. 28, No. 5, pp. 579-602, (1988).

[A12] J. W. Flower, S. W. Otto and M. C. Salama, "A Preprocessor for Irregular Finite Element Problems," *CalTech/JPL Report C3P-292,* (1986).

[A13] S. H. Bokhari, "On the Mapping Problem," *IEEE Trans. Comp.*, Vol. C-30, No. 3, pp. 207-214, (1981).

[A14] C. Farhat, "On the Mapping of Massively Parallel Processors Onto Finite Element Graphs," *Comp. & Struct.*, Vol. 32, No. 2, pp. 347-354, (1989).

[A15] S. Hammond and R. Schreiber, "Mapping Unstructured Grid Problems to the Connection Machine," *RIACS Technical Report 90.22,* (1990).

[A16] E. D. Dahl, "Mapping and Compiling Communication on the Connection Machine System," *Proc. Distr. Mem. Comp. Conf.*, Charleston, (1990).

[A17] M. Chrobak and D. Epstein, "Planar Orientations with Low Out-Degree and Compaction of Adjacency Matrices," *Theor. Comp. Sci., To appear,* (1990).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1994	Final Contractor Report

4. TITLE AND SUBTITLE

High-Performance Parallel Analysis of Coupled Problems for Aircraft Propulsion

5. FUNDING NUMBERS

WU–505–10–11
C–NAG3–1273

6. AUTHOR(S)

C.A. Felippa, C. Farhat, S. Lanteri, U. Gumaste, and M. Ronaghi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Colorado
Department of Aerospace Engineering Sciences and
Center for Space Structures and Controls Boulder, Colorado 80309–0429

8. PERFORMING ORGANIZATION REPORT NUMBER

E–8658
CU–CSSC–93–16

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR–195292

11. SUPPLEMENTARY NOTES

Project Manager, Christos C. Chamis, Structures Division, organization code 5200, NASA Lewis Research Center, (216) 433–3252.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 39

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

Applications are described of high-performance parallel, computation for the analysis of complete jet engines, considering its multi-discipline coupled problem. The coupled problem involves interaction of structures with gas dynamics, heat conduction and heat transfer in aircraft engines. The methodology issues addressed include: consistent discrete formulation of coupled problems with emphasis on coupling phenomena; effect of partitioning strategies, augmentation and temporal solution procedures; sensitivity of response to problem parameters; and methods for interfacing multiscale discretizations in different single fields. The computer implementation issues addressed include: parallel treatment of coupled systems; domain decomposition and mesh partitioning strategies; data representation in object-oriented form and mapping to hardware driven representation, and tradeoff studies between partitioning schemes and fully coupled treatment.

14. SUBJECT TERMS

Structural mechanics; Gas dynamics; Heat transfer multiscale; Decomposition sensitivity; Object oriented partitioning; Tradeoff

15. NUMBER OF PAGES
41

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	